Vous pouvez trouver notre projet sur:

https://www.youtube.com/watch?v=8oLDY_sgXjU

Barrière Automatique

01000010 01100001 01110010 01110010 01101001 01100101 01110010 01100101 00100000 01000001 01110101 01110100 01101111 01101101 01100001 01110100 01101001 01110001 01110101 01100101

66 97 114 114 105 101 114 101 32 65 117 116 111 109 97 116 105 113 117 101

PROJET 3

BARRIÈRE AUTOMATIQUE DE PARKING

Introduction du projet:

Nous allons présenter un projet de barrière automatique qui sert à controler un parking pour ne pas dépasser sa capacité de stationnement.
Les voies d'entrée et de sortie sont placés à deux endroits distincts.
L'entrée est conditionnée par l'existance d'une place disponible(feu vert:accès autorisé.feu rouge:accès interdit)
Le parking est équipé par un système de détection des véhicules entrants et sortants et d'un afficheur positionné à l'entrée de parking qui affiche le nombre de places disponibles.

On a besoin de:

-ouvrir et fermer l’accès du parking automatique.

-conditionner l’accès du parking par l’existance d’une place disponible.

-prévoir la possibilité de controler manuellement l’ouverture et la fermeture de la barrière à travers un bouton poussoir.

Solution:installer à l’entrée de parking une barrière qui s’ouvre et se ferme à travers d’une commande manuelle ou automatique.

II. Formalisation du besoin

Doc. 1 Diagramme bête à corne

Doc. 2 **Diagramme pieuvre**

FP : Fonction principle.

FC : Fonction complémentaire ou contrainte.

III. Élaboration du cahier des charges

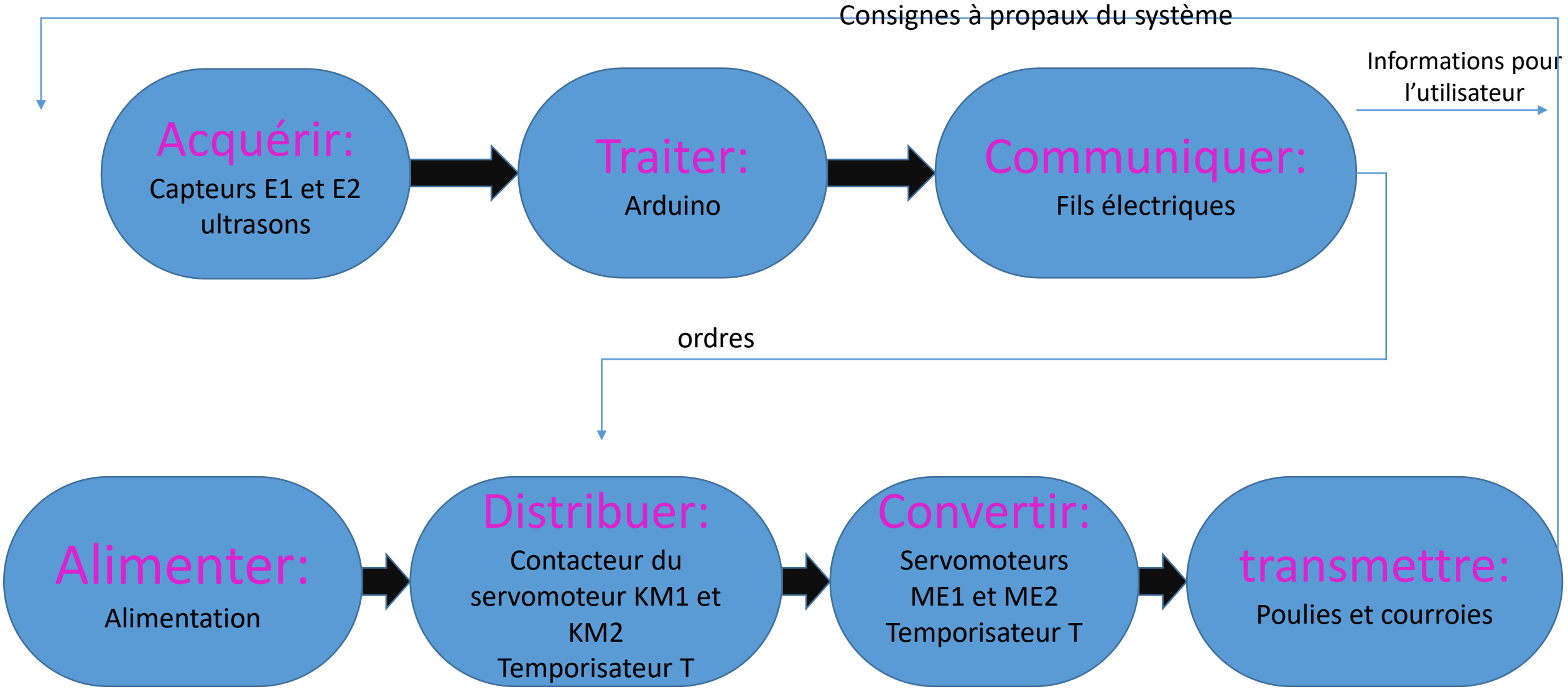
Doc. 3

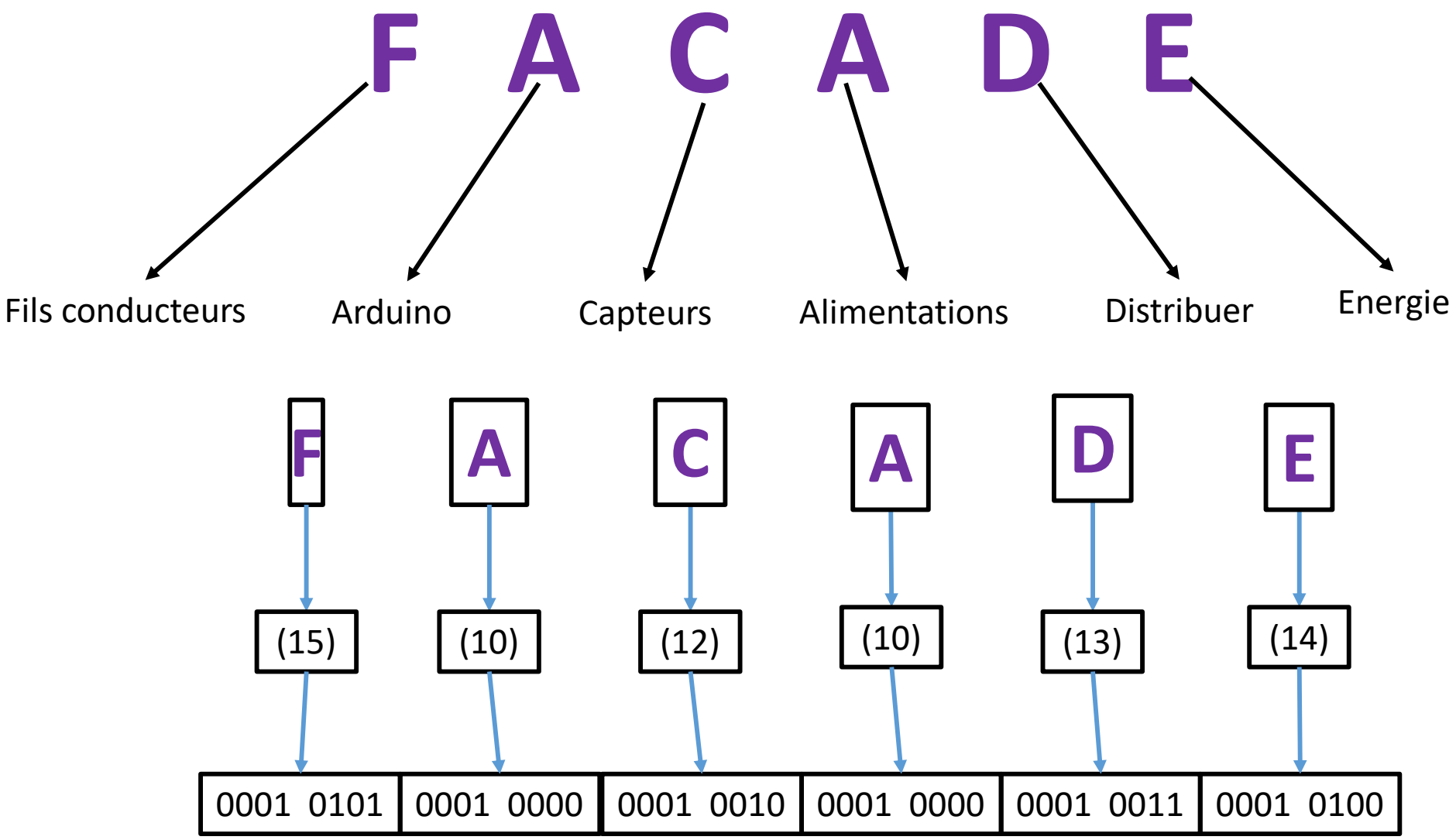
Cahier des charges fonctionnel (CdCF)

Rep.	Fonction	Critères	Niveau	Flexibilité
FP	Ouvrir et fermer l'accès du parking automatiquement en fonction des places disponibles.	• Ouverture • Fermeture • Places disponibles	• Instantanée • Après 3 s de la détection du passage du véhicule. • 10 places	• F1
FC1	S'adapter à l'accès du parking.	• Longueur • Largeur • Hauteur	• 15 cm ≤ L ≤ 20 cm • 5 cm ≤ l ≤ 10 cm • 5 cm ≤ h ≤ 10 cm	• F1 • F1 • F1
FC2	Être alimenté en énergie électrique.	• énergie	• Piles, batterie ou Énergies renouvelables	• F0
FC3	Respecter les règles de sécurité.	• Tension • Signalisation • Matériaux	• Tension ≤24v • Indications en rouge sur un fond blanc • Non toxiques	• F0 • F0 • F0
FC4	Faciliter l'installation et la réparation.	• Montage/ Démontage • Câblage	• Solutions simples et démontables • Accessible	• F0
FC5	Rentrer dans le budget du lycée.	• Prix de revient	• 100 dinars au maximum	• F1
FC6	Être réalisable au laboratoire	• Matériaux • Matériel	• Celui disponible au laboratoire	• F1
FC7	Respecter l'environnement	• Matériaux	• Recyclable	• F0

F0 : impératif - F1 : peu négociable - F2 : négociable - F3 : libre

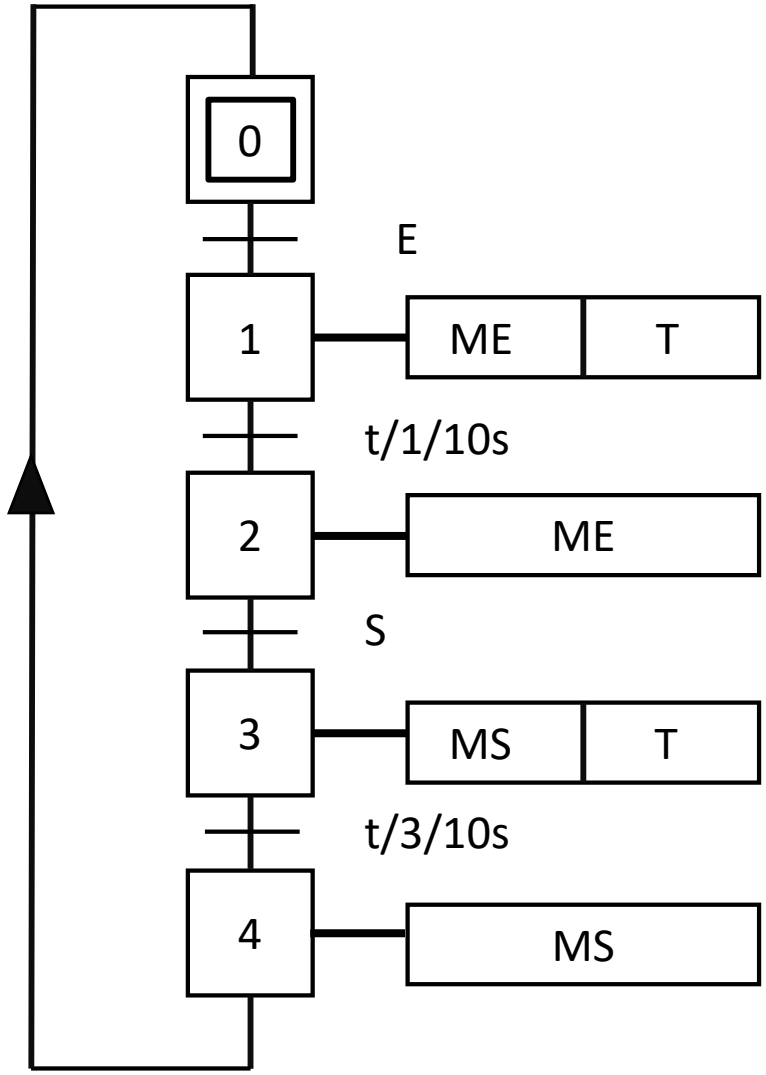
4)Chaine d'information et Chaine d'énergie:



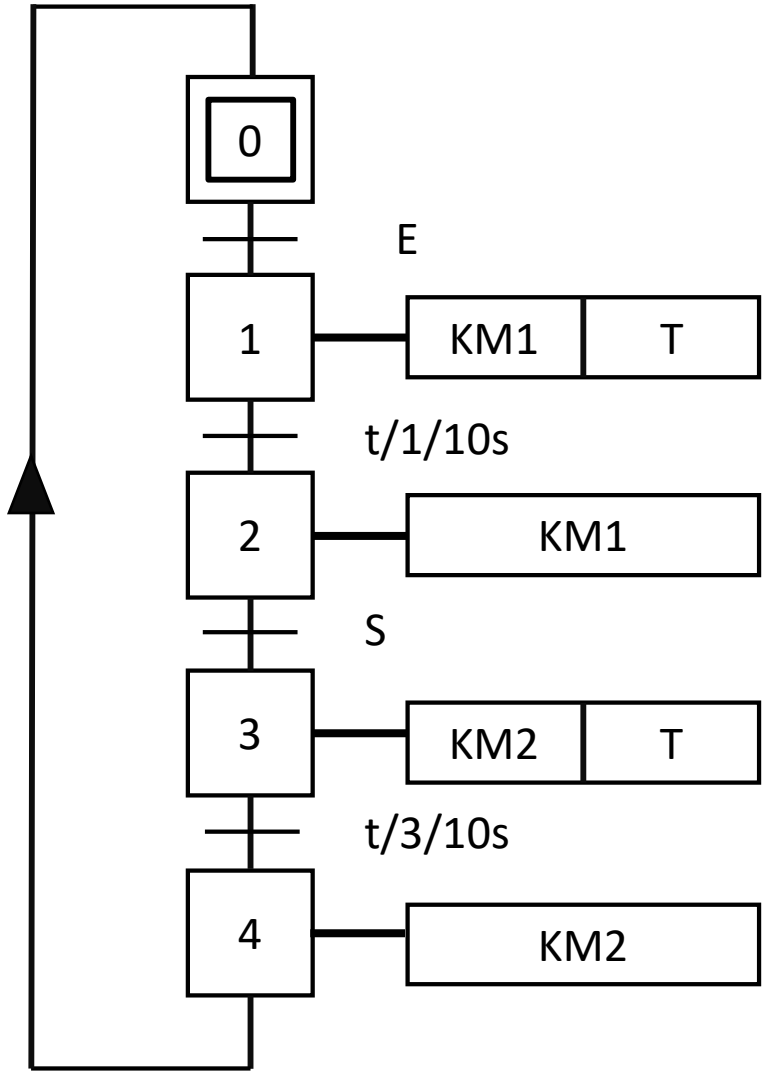

F A C A D E
Fils conducteurs
Arduino
Capteurs
Alimentations
Distribuer
Energie
F
A
C
A
D
E
(15)
(10)
(12)
(10)
(13)
(14)
0001 0101
0001 0000
0001 0010
0001 0000
0001 0011
0001 0100

Grafecet de point de vue de système :

GRAFFCET de point de vue opérative:

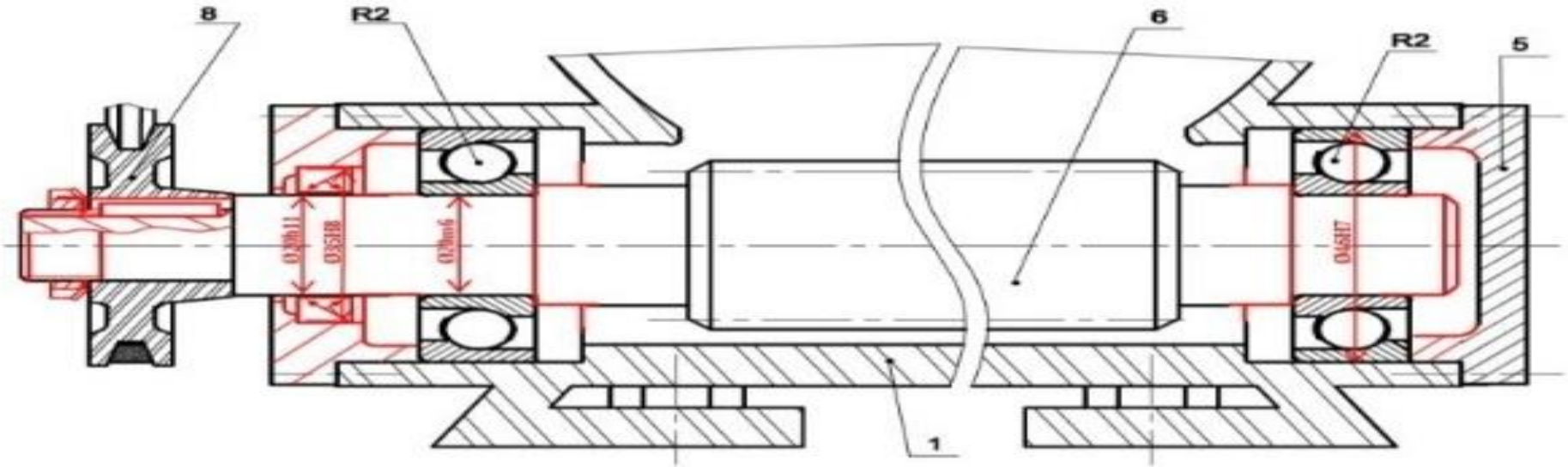


GRAFFCET de point de vue commande:



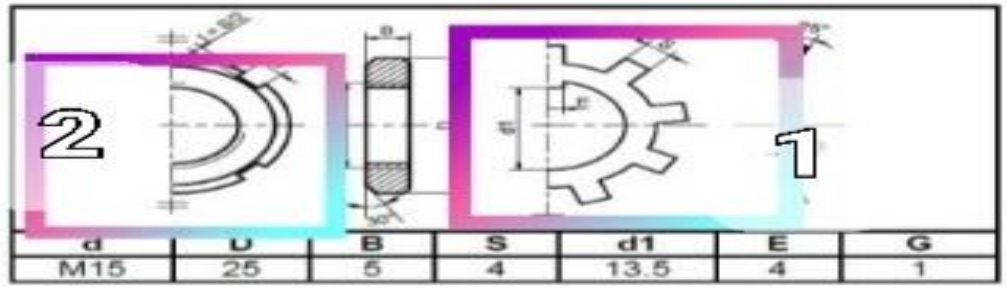
6)dessin d’ensemble:

NB:ce sont les tailles réelles du barrière automatique.



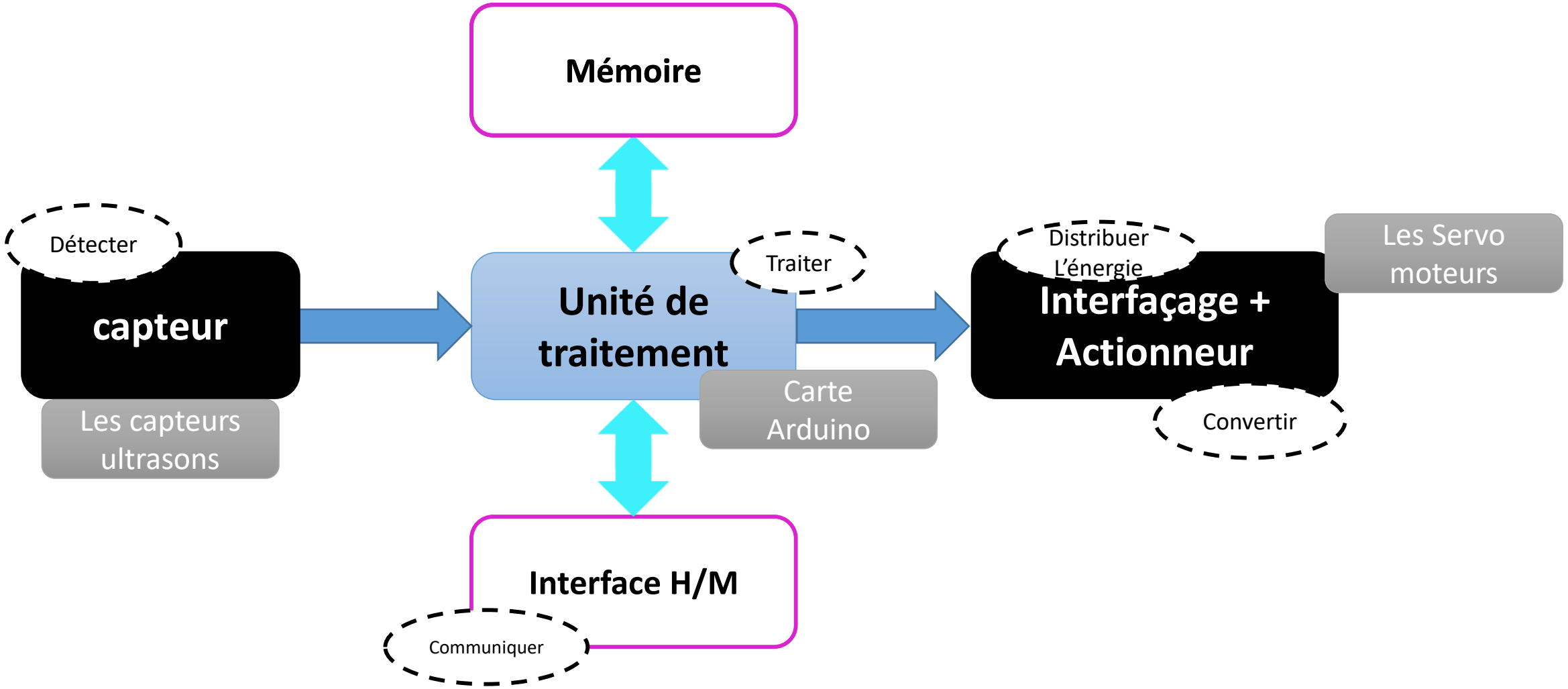
Joint à deux lèvres

d	D	E	d	D	E
17	28	7	20	35	7
18	32	7	22	40	7

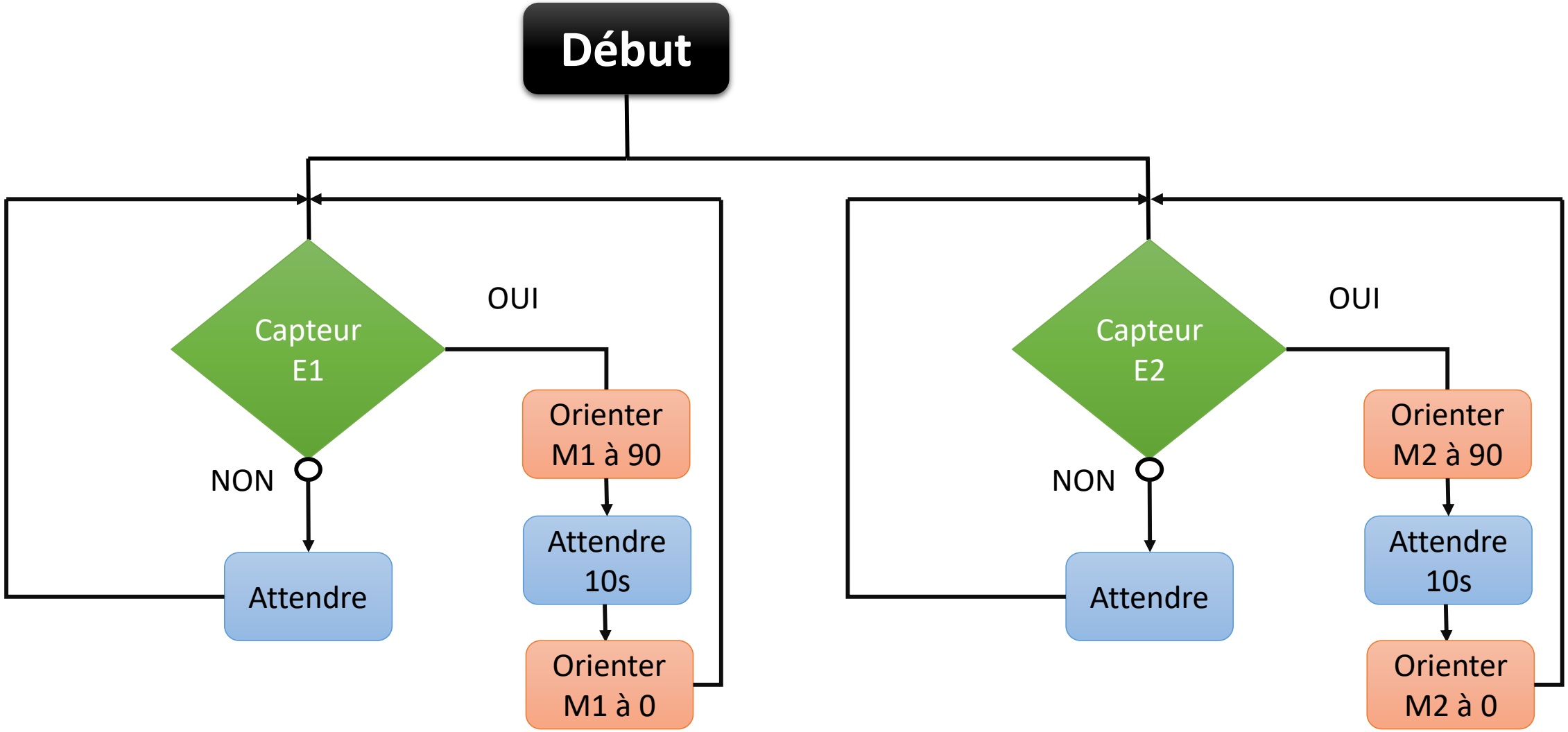


d	U	B	S	d1	E	G
M15	25	5	4	13.5	4	1

7)l'architecture du système:



8)l'organigramme du fonctionement:



9)Programmation

La carte Arduino ne supporte pas la langage des blocks, c'est pourquoi on doit convertir le code ci-dessus en la langage C. Cette transformation est appelée ***téléversement:***

Programme textuel

```
1  #include <Arduino.h> //librairie pour Arduino
2  #include <Wire.h> //librairie pour Arduino
3  #include <SoftwareSerial.h> //librairie pour Arduino
4  #include <Servo.h> //librairie pour servomoteur
5  double angle_rad = PI/180.0 ; // variable de position du servo en rad
6  double angle_deg = 180.0/PI ; // variable de position du servo en Deg
7  Servo servo_2 ; // crée un objet servo pour contrôler le servomoteur
8  Servo servo_4 ; // crée un objet servo pour contrôler le servomoteur
9  void setup () // Code d'initialisation
10 {
11     pinMode (3, INPUT) ; // Broche 3 du capteur IR configurée en entrée
12     servo_2. attach (2) ;// Broche servo_2 configurée en sortie
13     pinMode (5, INPUT) ; // Broche 5 du capteur IR configurée en entrée
14     servo_4. attach (4) ;// Broche servo_4 configurée en sortie
15 }
```

```
16  void loop () // Boucle sans fin (répéter)
17  {
18  If                //si
19  (((digitalRead (3)) == (1))) //état logique haut de la broche 3
20  {
21   servo_2. write (90) ; // orienter le servo_2 à 90°
22  _delay (10) ;// attendre 10s
23  }
24  Else // sinon
25  {
26  servo_2. write (0) ; // orienter le servo_2 à 0°
27  }
28  If
29  ((digitalRead (5)) == (1)))
30  {
31  servo_4. write (90);
32  _delay (10);
33  }
34  Else
35  {
36  servo_4. write (0);
37  }
38  _loop ();
39  }
40  void _delay (float seconds) {
41      long endTime = millis () + seconds * 1000;
42      while (millis () < endTime) _loop ();
43  }
44  void _loop () {
45  }
```

Exemple de fonctionnement:

https://www.youtube.com/watch?v=AX-LmCb6pYE

Les composants de notre projet: